

In the Claims:

Please cancel Claims 2-3 and 7-8; amend Claims 1, 5, 6 and 10; and add new Claims 11-23, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented claims in a continuing or future application.

1. (Currently Amended) A system for accessing ~~Java Message Service~~ an application program interface using a mark-up language, comprising:

a computer including a processing device operating thereon;

a source file stored on a computer readable medium, wherein the source file ~~that~~ contains one or more a plurality of markup language components commands;

a parser that executes on the computer and parses ~~said~~ the source file to extract the markup language commands, and ~~communicates said~~ communicate the markup language components commands to a command processor; ~~and;~~

a command processor that executes on the computer and validates the markup language commands, and, ~~converts the~~ for each markup language components command converts the markup language command into one of JMS or JMX system a command operations object for communication to a command dispatcher;

a command dispatcher that executes on the computer and receives command objects from the command processor and, for each command object, assigns the command object to one of a plurality of categories corresponding to a plurality of application program interfaces; and

a plurality of processor modules, including a processor module for each category of application program interface, wherein each processor module executes on the computer, receives the command objects assigned to its category, and performs appropriate operations against the corresponding application program interface.

2-3. (Canceled).

4. (Original) The system of claim 1 wherein the source file is an XML file.

5. (Currently Amended) The system of claim 4 wherein the markup language is JMS markup language, and wherein the XML file includes JMSML commands delimited within the XML file by beginning and ending JMSML tags.

6. (Currently Amended) A method for accessing ~~Java Message Service~~ an application program interface using a mark-up language, comprising the steps of:

supplying a source file, wherein said source file contains one or more markup language components;

passing said source file as input to a parser, wherein said parser identifies said markup language components and passes said markup language components to a command processor;
and;

translating, by said command processor, the markup language components into one of JMS or JMX system operations

retrieving a source file stored on a computer readable medium, wherein the source file contains a plurality of markup language commands;

parsing the source file to extract the markup language commands;

validating the markup language commands, and, for each markup language command converts the markup language command into a command object;

assigning each command object to one of a plurality of categories corresponding to a plurality of application program interfaces; and

processing the command objects using a plurality of processor modules, including a processor module for each category of application program interface, wherein each processor module receives the command objects assigned to its category, and performs appropriate operations against the corresponding application program interface.

7-8. (Canceled).

9. (Original) The method of claim 6 wherein the source file is an XML file.

10. (Currently Amended) The method of claim 9 wherein the markup language is JMS markup language, and wherein the XML file includes JMSML commands delimited within the XML file by beginning and ending JMSML tags.

11. (New) The system of claim 1 wherein at least one of the application program interfaces conforms to the Java Message Service specification.

12. (New) The system of claim 1 wherein at least one of the application program interfaces conforms to the Java Management Extensions specification.

13. (New) The system of claim 1 wherein the plurality of application program interfaces include both application program interface that conforms to the Java Message Service specification and an application program interface that conforms to the Java Management Extensions specification.

14. (New) The system of claim 1 wherein the source file further defines scenarios, and wherein each scenario specifies a group of multiple operations to be performed by the system over the plurality of application program interfaces, for a particular scenario.

15. (New) The method of claim 6 wherein at least one of the application program interfaces conforms to the Java Message Service specification.

16. (New) The method of claim 6 wherein at least one of the application program interfaces conforms to the Java Management Extensions specification.

17. (New) The method of claim 6 wherein the plurality of application program interfaces include both application program interface that conforms to the Java Message Service specification and an application program interface that conforms to the Java Management Extensions specification.

18. (New) The method of claim 6 wherein the source file further defines scenarios, and wherein each scenario specifies a group of multiple operations to be performed by the system over the plurality of application program interfaces, for a particular scenario.

19. (New) A computer readable medium including instructions stored thereon, which when executed cause the computer to perform the steps of:

- retrieving a source file stored on a computer readable medium, wherein the source file contains a plurality of markup language commands;

- parsing the source file to extract the markup language commands;

- validating the markup language commands, and, for each markup language command converts the markup language command into a command object;

- assigning each command object to one of a plurality of categories corresponding to a plurality of application program interfaces; and

- processing the command objects using a plurality of processor modules, including a processor module for each category of application program interface, wherein each processor module receives the command objects assigned to its category, and performs appropriate operations against the corresponding application program interface.

20. (New) The computer readable medium of claim 19 wherein at least one of the application program interfaces conforms to the Java Message Service specification.

21. (New) The computer readable medium of claim 19 wherein at least one of the application program interfaces conforms to the Java Management Extensions specification.

22. (New) The computer readable medium of claim 19 wherein the plurality of application program interfaces include both application program interface that conforms to the Java Message Service specification and an application program interface that conforms to the Java Management Extensions specification.

23. (New) The computer readable medium of claim 19 wherein the source file further defines scenarios, and wherein each scenario specifies a group of multiple operations to be performed by the system over the plurality of application program interfaces, for a particular scenario.